

## Research Article

# Simultaneous Topology, Shape, and Sizing Optimisation of Plane Trusses with Adaptive Ground Finite Elements Using MOEAs

Norapat Noilublao<sup>1</sup> and Sujin Bureerat<sup>2</sup>

<sup>1</sup> Department of Engineering Management, Faculty of Science and Technology, Rajabhat Maha Sarakham University, Maha Sarakham 44000, Thailand

<sup>2</sup> Department of Mechanical Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand

Correspondence should be addressed to Sujin Bureerat; [sujbur@kku.ac.th](mailto:sujbur@kku.ac.th)

Received 4 October 2012; Revised 7 January 2013; Accepted 8 January 2013

Academic Editor: Sheng-yong Chen

Copyright © 2013 N. Noilublao and S. Bureerat. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a novel integrated design strategy to accomplish simultaneous topology shape and sizing optimisation of a two-dimensional (2D) truss. An optimisation problem is posed to find a structural topology, shape, and element sizes of the truss such that two objective functions, mass and compliance, are minimised. Design constraints include stress, buckling, and compliance. The procedure for an adaptive ground elements approach is proposed and its encoding/decoding process is detailed. Two sets of design variables defining truss layout, shape, and element sizes at the same time are applied. A number of multiobjective evolutionary algorithms (MOEAs) are implemented to solve the design problem. Comparative performance based on a hypervolume indicator shows that multiobjective population-based incremental learning (PBIL) is the best performer. Optimising three design variable types simultaneously is more efficient and effective.

## 1. Introduction

Nowadays, the use of artificial intelligence techniques in a wide variety of engineering applications has become commonplace as a means of responding to needs to deal with complicated systems which classical mathematical methods cannot solve [1, 2]. Such techniques include, for example, neural networks, fuzzy sets, and evolutionary computation [1]. Applications of evolutionary algorithms (EAs) for tackling engineering design problems have become increasingly popular due to certain advantages. The methods are simple to use, robust, and capable of dealing with almost any kind of design problem, for example, structural design [3–8], supply chain management [9], quay management [10], sensor placement [11], and manufacturing tasks [12, 13]. Design variables can be discrete, continuous, and combinatorial. They can search for global optima effectively. More attractively, the multiobjective versions of evolutionary algorithms can be used to explore a Pareto front within one optimisation run. Nevertheless, EAs have some unavoidable drawbacks as they

have a low convergence rate and a lack of consistency due to randomisation being used in a search process. As a result, there are always needs to improve EAs performance. The hybridisation of existing methods with local search techniques [14] or other EA concepts is a means to achieve such a goal [12, 15]. Also, the use of surrogate models for design problems with expensive function evaluations is an effective way to enhance EAs [16]. When using EAs to solve a new type of design problem, it is always useful to compare the search performance of a number of EAs for such a problem. This is because one particular EA may be efficient for one design problem but it is unlikely to be the best performer for other types of problems [17].

A truss structure is one of the most used structures in engineering applications. Using such a structure is said to be advantageous, since they are simple and inexpensive to construct. It can be employed for many engineering purposes, for example, a billboard structure, a factory roof structure, a bridge structure, and a wind turbine tower. In the past, the design of such a structure was usually carried out in such

a way that an initial structural configuration was formed by an experienced engineering designer for conceptual design. The classical civil engineering design approach is then applied in the preliminary and detailed design stages. In recent years, considerable research work towards design/optimisation of trusses and frames has been conducted. Thousands of research papers have been published (e.g., [18–28]). For truss optimisation, design variables can be classified as topological [23, 29–33], shape, and sizing variables [5, 7, 34–38]. In the past, a designer traditionally performs topology optimisation in the first design phase to obtain an initial structural layout. Shape and sizing optimisation are then conducted to further improve the structure. This can be called a multilevel or multistage optimisation process. Nevertheless, it has been investigated that the better design process is to perform topology, shape, and sizing optimisation simultaneously [28]. Combined topology and sizing [21, 24], shape and sizing [34–38], topology and shape [18, 20, 22], and all three variable types [19, 26, 28, 39–41] have been recently investigated.

One of the most effective design strategies for truss topology optimisation is the use of a ground element approach. The topological design process based upon ground elements can be tackled by both gradient-based [21, 31, 42] and population-based optimisers [28, 29, 33]. The gradient-based methods have advantages in that they have high convergence rate and search consistency (obtaining the same design solution when solving the same problem several times). They are powerful for a large-scale topology design problem. Nevertheless, the population-based or evolutionary approaches have other advantages, since they are more robust and do not require function derivatives for searching. Moreover, for multiobjective design cases, evolutionary algorithms have a special feature: to explore a nondominated front within one optimisation run. This, together with the capability of solving all kinds of design problems, makes the evolutionary algorithms popular among researchers and designers.

This paper presents a novel design strategy to achieve simultaneous topology shape and sizing design of a planar truss structure. A multiobjective design problem with two different sets of design variables is assigned in this work. The problem is minimising structural mass and compliance subject to stress, buckling, and compliance constraints. The first set of design variables is used to design structural topology and truss element sizing while structural shape variables are added to the second set of design variables. These two design variable sets in combination with the adaptive ground element concept result in two multiobjective design problems. MOEAs employed to tackle the problems are second version of strength Pareto evolutionary algorithm (SPEA) [43], population-based incremental learning [44], archived multiobjective simulated annealing (AMOS) [45], multiobjective particle swarm optimisation (MPSO) [46], and unrestricted population size evolutionary multiobjective optimisation algorithm (UPS-EMOA) [47]. The results obtained from using the various optimisers are compared and discussed. It is shown that the second design variables set which includes all types of design variables is superior to the first set.

The rest of the paper is organised as follows. Section 2 briefly details MOEAs. A multiobjective design problem is

expressed in Section 3. Section 4 details the decoding/encoding procedures for simultaneous topology, shape, and sizing design variables with adaptive ground elements. Design results and comparative performance are shown in Section 5 while conclusions are drawn in Section 6.

## 2. Multiobjective Evolutionary Algorithms

The field of multiobjective evolutionary optimisation is one of the hottest issues in evolutionary computation. The most outstanding ability of MOEAs is that they can explore a Pareto front within one simulation run. This in combination with their simplicity, robustness, and capability of dealing with all kind of variables makes this kind of optimiser more popular and attractive. MOEAs search mechanisms are based on a population of design solutions, which work in such a way that a population is evolved iteration by iteration. A matrix called an external Pareto archive is used to collect nondominated solutions iteratively. The Pareto archive is updated until a termination criterion is fulfilled. The MOEAs used for a comparative performance test in this paper are given below.

*2.1. Strength Pareto Evolutionary Algorithm.* SPEA was proposed by Zitzler and Thiele [48], and later its improved version SPEA2 [43]. The search procedure starts with an initial population and an external Pareto set. Fitness values are assigned to the population based upon the levels of domination and crowding. A set of solutions are then selected to a mating pool by means of a binary tournament selection operator. A new population is produced using crossover and mutation on those selected individuals. The updated external Pareto solutions are the nondominated solutions of the union set of the previous external Pareto set and the new population. In cases that the Pareto archive is full, the nearest neighbourhood technique is invoked to remove some design solutions from the archive. The Pareto archive is updated repeatedly until the termination criterion is met.

*2.2. Population-Based Incremental Learning.* The algorithm of multiobjective PBIL [44] starts with an external Pareto archive and initial probability matrix having all elements set to be 0.5. A set of binary design solutions are then created corresponding to the probability matrix while their function values are evaluated. The Pareto archive is updated with the nondominated solutions of the new population and the members of the previous Pareto archive. In cases that the number of nondominated solutions exceeds the predefined archive size, the normal line method is activated to remove some members from the archive. The probability matrix and the Pareto archive are iteratively updated until the termination criteria are fulfilled.

*2.3. Archived Multiobjective Simulated Annealing.* Simulated Annealing (SA) is one of the most popular random-directed optimisers. It has long been used in a wide variety of design applications [49, 50]. The method is based upon mimicking the random behaviour of molecules during the annealing

process, which involves slow cooling from a high temperature. For AMOSA [45], the method starts with initial temperature, a population, and a Pareto archive. A simple downhill technique is then applied to improve the population and update the archive for a few iterations. Then, a parent design solution is randomly chosen from the archive and it is used to generate a child or candidate solution by means of mutation. The parent is replaced by its offspring if the offspring dominates it. In cases that the parent is nondominated by the offspring, the parent still has a probability of being replaced. Such a probability is based on Boltzmann probability and the amount of domination between them. As the children are created, the Pareto archive is updated while the annealing temperature is reduced iteratively until a termination condition is met. An archiving technique for this algorithm is called a clustering technique.

**2.4. Multiobjective Particle Swarm Optimisation.** The particle swarm optimisation method uses real codes and searches for an optimum by mimicking the movement of a flock of birds, which aim to find food [46]. The search procedure herein is based on the particle swarm concepts combined with the use of an external Pareto archiving scheme. Starting with an initial set of design solutions (viewed as particles) as well as their initial velocities and objective function values, an initial Pareto archive is filled with the nondominated solutions obtained from sorting the initial population. A new population is then created by using the particle swarm updating strategy where the global best solution is randomly selected from the external Pareto archive. Afterwards, the external archive is updated by the nondominated solutions of the union set of the new population and the previous nondominated solutions. In cases that the number of nondominated solutions is too large, the adaptive grid algorithm [51] is employed to properly remove some solutions from the archive. The Pareto archive is repeatedly improved until fulfilling the termination criteria.

**2.5. Unrestricted Population Size Evolutionary Multiobjective Optimisation Algorithm.** The unrestricted population-size EMO algorithm was proposed by Aittokoski and Miettinen [47]. Similarly to the nondominated sorting genetic algorithm (NSGAI) [52], this optimiser uses a population to contain nondominated solutions, but its population size is unlimited. Some solutions from an initial population are randomly selected to produce some offspring by using a differential evolution operator. A new population contains nondominated solutions sorted from the combination of the offspring and the current population. The method has an unrestricted population size, therefore this somewhat provides population diversity. The optimiser is usually terminated with the maximum number of function evaluations.

### 3. Topological Design

Structural topology optimisation of trusses using the ground element approach can be performed in such a way that

ground finite elements, all possible combinations of predefined nodes in the given design domain, are created in the initial stage. Topological design variables determine elements' cross-sectional areas. After an optimisation run, elements with very small sizes will be removed from the structure while other elements are retained as truss members. With such a concept, an initial structural configuration is achieved. In this work, the biobjective design problem for a 2D truss with simultaneous topology shape and sizing design variables can be expressed as follows:

$$\begin{aligned} \min \quad & \{\text{mass}, c\}, \\ \text{subject to} \quad & \sigma_{\max} \leq \sigma_{\text{al}}, \\ & \lambda_i \leq 1, \\ & c \leq c_{\text{al}}, \\ & \mathbf{x} \in \Omega, \end{aligned} \quad (1)$$

where mass is structural mass,  $c$  is structural compliance,  $c_{\text{al}}$  is allowable compliance, and  $\sigma_{\max}$  is the maximum stress on truss elements.  $\sigma_{\text{al}} = \sigma_{yt}/N_F$  is an allowable stress,  $\lambda_i$  is a buckling factor for each element (defined as the ratio of applied load to critical load),  $\sigma_{yt}$  is yield strength,  $N_F$  is the factor for safety,  $\mathbf{x}$  is a vector of design variables, and  $\Omega$  is a design domain of  $\mathbf{x}$ .

The first design objective, structural mass, is set to minimise cost while minimising the second objective, which is equivalent to structural stiffness maximisation. The stress and buckling constraints are assigned for safety requirements while the compliance constraint is imposed so as to obtain reasonable structural layouts. For one evaluation, with the input design variables, a finite element (FE) model of a truss is obtained and the FE analysis is performed. Having obtained nodal displacement and axial stresses of truss elements, the local buckling factor for each element can then be computed.

### 4. Adaptive Ground Elements

Two design variables sets are assigned in this paper. The first set is created for simultaneous topology and sizing optimisation. Figure 1 shows a rectangular design domain sized  $W \times H$  for a 2D truss structure used in this study. The structure is subjected to point load  $F = 500$  N at the top right-hand corner of the domain [33]. The width was set to be constant at 2 meters while the domain height could be varied in the range of  $[H_{\min}, H_{\max}] = [0.25, 1]$  meter. Fixed boundary conditions were assigned at the left edge of the domain as shown.

Given that an array of predefined nodes is defined, ground elements as the combinations between the nodes are formed as shown in Figure 2. With the parameters  $n_x$  and  $n_y$  being the numbers of equispaced nodes in the  $x$  and  $y$  directions, respectively,  $n_x \times n_y$  nodal positions were then generated where, in this study,  $n_x \in \{3, 4, 5\}$  and  $n_y \in \{2, 3, 4\}$ . The parameters  $n_x$  and  $n_y$  are the numbers of rows and columns of an input node array, respectively. The total number of ground elements as  $N_e = (n_x - 1)n_y + 3(n_x - 1)(n_y - 1)$  node combinations could be obtained as shown in Figure 2.

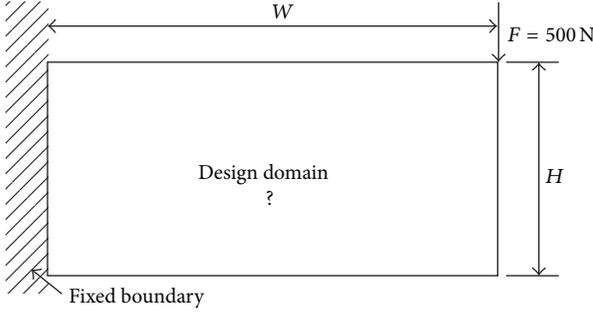


FIGURE 1: Design domain.

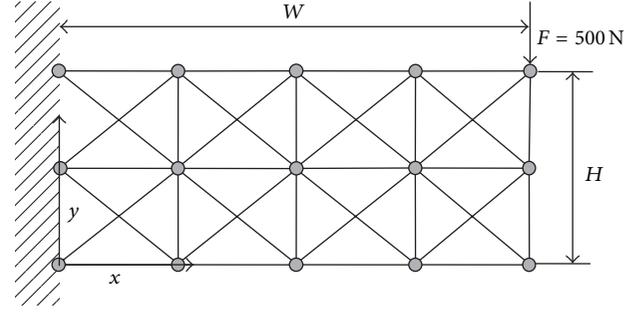


FIGURE 2: Adaptive ground elements.

The design variables include the parameters  $H$ ,  $n_x$ , and  $n_y$  and the diameters of the ground elements. Given that  $n_{x,\max}$  and  $n_{y,\max}$  are the maximum values for  $n_x$  and  $n_y$ , respectively, the possible maximum number of design variables is set as  $N_{e,\max} + 3$  where  $N_{e,\max} = (n_{x,\max} - 1)n_{y,\max} + 3(n_{x,\max} - 1)(n_{y,\max} - 1)$ . The computational steps for design variables decoding were as follows.

*Input.*  $\mathbf{x}$ , sized  $(N_{e,\max} + 3) \times 1$ ;  $x_i \in [0, 1]$ .

- (1) Set  $H = H_{\min} + x_1(H_{\max} - H_{\min})$ ,  $n_x = \text{round}(n_{x,\min} + x_2(n_{x,\max} - n_{x,\min}))$ , and  $n_y = \text{round}(n_{y,\min} + x_3(n_{y,\max} - n_{y,\min}))$ .
- (2) Compute  $N_e = (n_x - 1)n_y + 3(n_x - 1)(n_y - 1)$ .
- (3) Generate  $n_x \times n_y$  structural nodes, and  $N_e$  ground elements.
- (4) Assign element diameters. Set  $y_j = 4x_i$  for  $i = 4, \dots, N_e + 3$  and  $j = 1, \dots, N_e$ , which means  $y_j \in [0, 4]$ .
- (5) For  $j = 1$  to  $N_e$ 
  - (5.1) If  $\text{round}(y_j) = 0$ , set  $d_j = 0.000001$  m
  - (5.2) If  $\text{round}(y_j) = 1$ , set  $d_j = 0.01$  m
  - (5.3) If  $\text{round}(y_j) = 2$ , set  $d_j = 0.02$  m
  - (5.4) If  $\text{round}(y_j) = 3$ , set  $d_j = 0.03$  m
  - (5.5) If  $\text{round}(y_j) = 4$ , set  $d_j = 0.04$  m
- End
- (6) Assign fixed boundary conditions for all nodes at  $x = 0$ .
- (7) Apply the  $y$ -direction point load at the last node number.

*Output.* Nodal positions, ground element connections, boundary conditions, force vector, and elements' diameters.

The function  $\text{round}(x)$  gives the nearest integer to  $x$ . Although the design vector is sized  $(N_{e,\max} + 3) \times 1$ , only  $N_e + 3$  elements were used for each evaluation. The design variables are said to be mixed integer/continuous, which are simpler to deal with by using MOEAs. After having the nondominated solutions, elements with 0.000001 m diameter (or the case of

round ( $y_j$ ) = 0) were deleted from the ground structure to form a structural layout. A small element diameter is assigned in order to prevent singularity in a global stiffness matrix of a finite element model.

The second design variables set is an extension of the first set where parameters for nodal positions are added. For this study, initial positions of the truss joint are set as an array as the first design variables set shown in Figure 2. Additional shape design parameters determine the position changes of those nodes in  $x$ - and  $y$ -directions. All truss nodal positions except the fixed joints and the top row nodes are assigned to be varied in both  $x$ - and  $y$ -directions during an optimisation process. The top row nodal positions are allowed to be changed in the  $x$  direction in order to keep all top row nodes at the same level. As a result, the number of additional shape variables is  $N_{sh} = (n_x - 1)(2n_y - 1)$ . Therefore, the possible maximum number of design variables for this case is  $N_{e,\max} + N_{sh,\max} + 3$  where  $N_{sh,\max} = (n_{x,\max} - 1)(2n_{y,\max} - 1)$ . The computational steps for design vector encoding can be given as follows.

*Input.*  $\mathbf{x}$ , sized  $(N_{e,\max} + N_{sh,\max} + 3) \times 1$ ;  $x_i \in [0, 1]$ .

- (1) Set  $H = H_{\min} + x_1(H_{\max} - H_{\min})$ ,  $n_x = \text{round}(n_{x,\min} + x_2(n_{x,\max} - n_{x,\min}))$ , and  $n_y = \text{round}(n_{y,\min} + x_3(n_{y,\max} - n_{y,\min}))$ .
- (2) Compute  $N_e = (n_x - 1)n_y + 3(n_x - 1)(n_y - 1)$ .
- (3) Compute  $N_{sh,x} = n_y(n_x - 1)$  and  $N_{sh,y} = (n_x - 1)(n_y - 1)$ .
- (4) Compute the move limit for  $x$ -direction nodal position as  $\Delta x_{\min} = -(W/2/n_{x,\max} - 0.01)$ , and  $\Delta x_{\max} = W/2/n_{x,\max} - 0.01$  meter.
- (5) Compute the move limit for  $y$ -direction nodal position as  $\Delta y_{\min} = -(H_{\min}/2/n_{y,\max} - 0.01)$ , and  $\Delta y_{\max} = H_{\min}/2/n_{y,\max} - 0.01$  meter.
- (6) Generate  $n_x \times n_y$  initial structural nodes  $\mathbf{X}$ ,  $\mathbf{Y}$ , and  $N_e$  ground elements where  $\mathbf{X}$  and  $\mathbf{Y}$  are nodal coordinates in  $x$ - and  $y$ -directions, respectively.
- (7) Assign element diameters. Set  $y_j = 4x_i$  for  $i = 4, \dots, N_e + 3$  and  $j = 1, \dots, N_e$ .
- (8) For  $j = 1$  to  $N_e$

- (8.1) If  $\text{round}(y_j) = 0$ , set  $d_j = 0.000001$  m
- (8.2) If  $\text{round}(y_j) = 1$ , set  $d_j = 0.01$  m
- (8.3) If  $\text{round}(y_j) = 2$ , set  $d_j = 0.02$  m
- (8.4) If  $\text{round}(y_j) = 3$ , set  $d_j = 0.03$  m
- (8.5) If  $\text{round}(y_j) = 4$ , set  $d_j = 0.04$  m
- End
- (9) Set  $\Delta x_j = \Delta x_{\min} + (\Delta x_{\max} - \Delta x_{\min})x_i$  for  $i = (N_e + 4), \dots, (N_e + N_{sh,x} + 3)$  and  $j = 1, \dots, N_{sh,x}$ . Update  $x$ -direction nodal positions  $X_j = X_j + \Delta x_j$  for all node positions allowed to be changed.
- (10) Set  $\Delta y_j = \Delta y_{\min} + (\Delta y_{\max} - \Delta y_{\min})x_i$  for  $i = (N_e + N_{sh,x} + 4), \dots, (N_e + N_{sh,x} + N_{sh,y} + 3)$  and  $j = 1, \dots, N_{sh,y}$ . Update  $y$ -direction nodal positions  $Y_j = Y_j + \Delta y_j$  for all node positions allowed to be changed.
- (11) Assign fixed boundary condition for all nodes at  $x = 0$ .
- (12) Apply the  $y$ -direction point load at the last node number.

*Output.* Nodal positions, ground element connections, boundary conditions, force vector, and elements' diameters.

The design vector is sized  $(N_{e,\max} + N_{sh,\max} + 3) \times 1$ ; however, only  $N_e + N_{sh} + 3$  elements were used for each finite element model. The design problem (1) using the first set of design variables is termed OPT1 whereas the problem using the second set of design variables is named OPT2. The structure is made up of a material with Young modulus, yield strength, and density of  $214 \times 10^9$  Pa,  $360 \times 10^6$  Pa, and  $7850 \text{ kg/m}^3$ , respectively. The factor for safety was set to be 1.5 while the parameter  $c_{al}$  was set to be 0.04 N-m. Seven multiobjective evolutionary strategies are employed in this study where the optimisation settings are detailed as follows.

SPEA1: version two of SPEA using real codes with crossover and mutation rates of 1.0 and 0.05, respectively.

SPEA2: version two of SPEA using real codes with crossover and mutation rates of 1.0 and 0.5, respectively.

SPEA3: version two of SPEA using real codes with crossover and mutation rates of 1.0 and 0.75, respectively.

PBIL using binary codes with 0.05 mutation rate: and 0.2 mutation shift.

AMOSAs: performs a simple downhill operator for the first 3 generations before entering the main procedure. In this paper, the method uses real codes where a candidate is obtained by means of mutation.

MPSO: using real codes with a starting inertia weight, an ending inertia weight, a cognitive learning factor, and a social learning factor of 0.75, 0.1, 0.75, and 0.75, respectively.

TABLE 1: Hypervolume comparison of OPT1 and OPT2.

Optimiser	OPT1		OPT2	
	Mean	Std	Mean	Std
SPEA1	0.7098	0.0998	0.8619	0.0267
SPEA2	0.8161	<b>0.0343</b>	0.9274	<b>0.0184</b>
SPEA3	0.8400	0.0564	0.9236	0.0219
PBIL	<b>0.9330</b>	0.0647	<b>0.9519</b>	0.0527
AMOSAs	0.6619	0.1143	0.7997	0.0277
MPSO	0.3043	0.1371	0.6699	0.0582
UPSEMOA	0.4334	0.2808	0.6136	0.3885

UPSEMOA: using crossover probability, scaling factor, probability of choosing element from offspring in crossover, minimum population size, and burst size of 0.7, 0.8, 0.5, 10, and 25, respectively.

SPEA and PBIL are the best optimisers in the previous studies [28] while AMOSAs is selected instead of the usual NSGAI. MPSO is an optimiser with a different concept from the aforementioned MOEAs while UPSEMOA is used as it is a newly developed algorithm. For OPT1, the number of generations of MOEAs is set to be 250 while both the population and archive size are set as 150. For the second test problem, the number of generations of MOEAs is set to be 300 while both the population and archive size are set as 200. Each optimiser is applied to solve each design problem for 10 optimisation runs starting with the same initial population. The last updated Pareto archive is considered a Pareto optimal set. The nondominated sorting scheme presented in [53] is used to cope with design constraints.

## 5. Design Results

Having performed each multiobjective evolutionary strategy solving each design problem for ten simulation runs, the performance comparison of the various MOEAs is investigated based on a hypervolume indicator. The hypervolume indicator is one of the most reliable performance indicators used to examine the quality of approximate Pareto fronts obtained from using MOEAs. The parameter determines the area for biobjective or volume for more than two objectives covered by a particular nondominated front with respect to a given reference point. Since there are ten fronts for each method and for each design problem, the average of ten hypervolume values is employed to measure the convergence rate of the optimiser whereas the standard deviation is used to indicate search consistency.

The comparative hypervolumes of the various MOEAs for solving OPT1 and OPT2 are given in Table 1. Note that the hypervolumes in the table are normalised for ease in comparing. It is shown that PBIL is the best optimiser for both design cases based upon the convergence rate while SPEA2 is the most consistent method. However, as the standard deviation of SPEA2 does not approach to zero, the convergence rate is the most important indicator for this study.

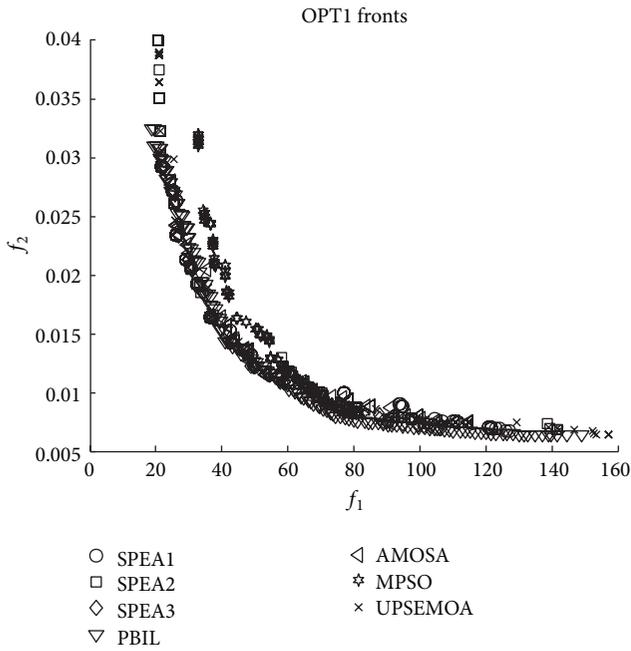


FIGURE 3: OPT1 best Pareto fronts from MOEAs.

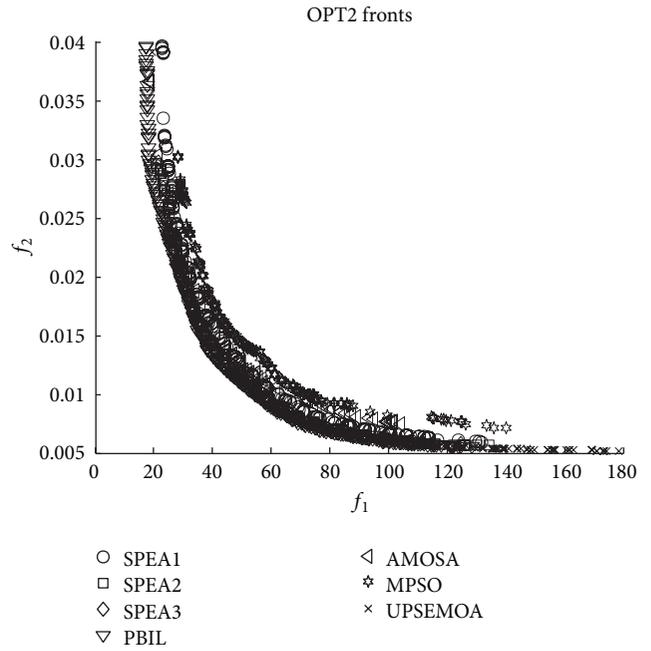


FIGURE 5: OPT2 best Pareto fronts from MOEAs.

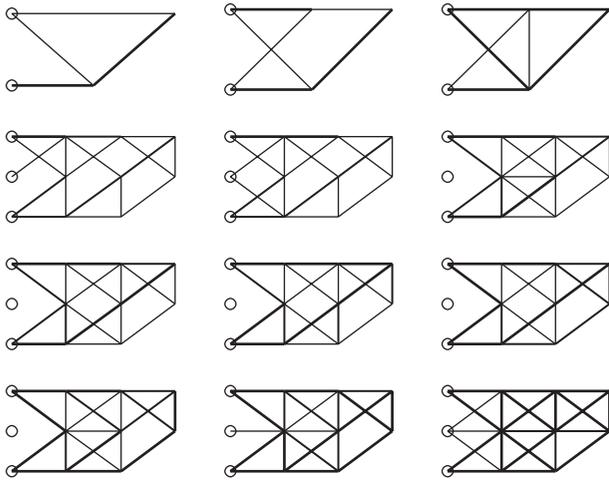


FIGURE 4: Some structures from the best front of PBIL in Figure 3.

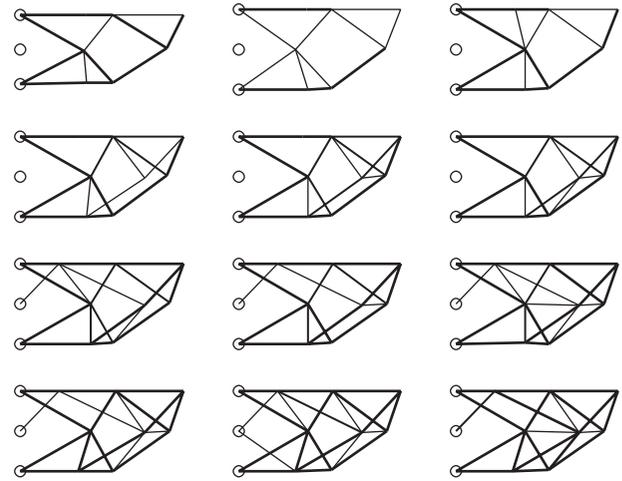


FIGURE 6: Some structures from the best front of PBIL in Figure 5.

Among the SPEA versions, it can be said that SPEA with higher mutation rate is superior to the low mutation rate version. SPEA3 is the second best optimiser for OPT1 while the worst performer for this problem is MPSO. For OPT2, the second best is SPEA2 closely followed by SPEA3 as the third best. The worst for this design case is UPSEMOA.

For the OPT1 design problem, the best fronts obtained from the best run of each multiobjective optimiser are plotted in Figure 3. It can be seen that the fronts are close together although the PBIL front has slightly more advancement and extension. This implies that MOEAs are efficient for this design problem. Some trusses selected from the PBIL front are displayed in Figure 4. The structures have a variety of structural layouts and elements' sizes; however, there are only two sets of node number, that is,  $2 \times 1$  and  $4 \times 2$ .

Figure 5 shows Pareto fronts obtained from the best runs of the various optimisers for the OPT2 case. Similarly to the first design case, all seven fronts are somewhat close together with the front obtained from using PBIL having slightly more advancement and extension. Some selected solutions from the PBIL front are displayed in Figure 6. The structures for this case have various topologies, and element dimensions, but tend to have fairly similar shapes. Changing nodal position has significant impact on the resulting structures. There is only one set of nodes array for the ground elements, which is  $4 \times 2$ . The best fronts obtained from solving OPT1 and OPT2 are plotted together for comparison in Figure 7. It is shown that the best front of OPT2 totally dominates that of OPT1. This implies that the second set of design variables is more efficient and effective.

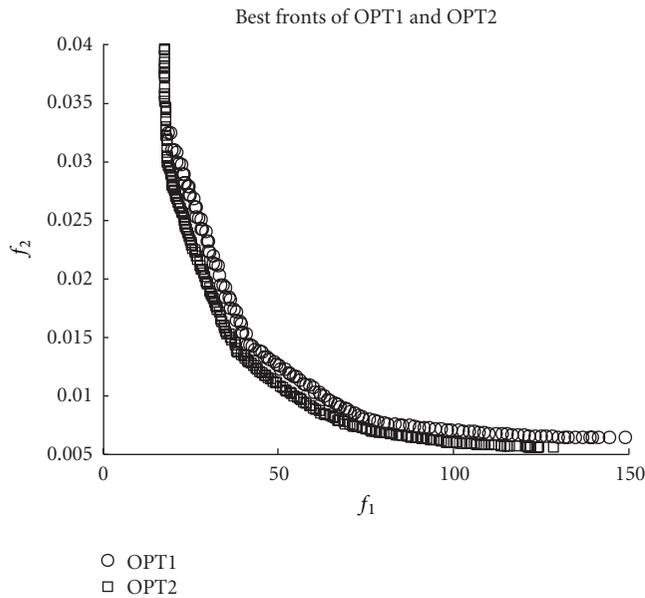


FIGURE 7: OPT1 front versus OPT2 front.

## 6. Conclusions and Discussion

The design results show that the proposed design strategies are efficient and effective as a variety of reasonable structural configurations could be obtained within one optimisation run. The obtained structures are said to be ready to use as they satisfy all the constraints for safety requirements. The multiobjective topological, shape, and sizing design with an adaptive ground element approach proposed herein can be tackled by an efficient multiobjective evolutionary algorithm. Among the MOEAs employed in this paper, PBIL is the most powerful method for solving both design problems based on the hypervolume indicator. For SPEA as the second best optimiser, using a higher mutation rate is more efficient. The second set of design variables which combines all types of design variables at the same time is superior to the first set which uses only topology and sizing variables. The proposed design approach can be a powerful engineering design tool in the future since, with one optimisation run, various feasible truss structures are obtained for decision making.

Future work could be the application of this design concept to a large-scale truss design problem. Alternative adaptive ground element formations can be created. The application to three-dimensional practical trusses and frames is challenging. Also, performance enhancement of multiobjective optimisers for solving this design problem is to be conducted.

## Acknowledgments

The authors are grateful for the support from the Thailand Research Fund (TRF), BRG5580017, and the Graduate School, Rajabhat Maha Sarakham University, Thailand.

## References

- [1] P. Lu, S. Chen, and Y. Zheng, "Artificial intelligence in civil engineering," *Mathematical Problems in Engineering*, vol. 2012, Article ID 145974, 22 pages, 2012.
- [2] C. Cattani, S. Chen, and G. Aldashev, "Information and modeling in complexity," *Mathematical Problems in Engineering*, vol. 2012, Article ID 868413, 3 pages, 2012.
- [3] A. R. Yildiz, "Comparison of evolutionary-based optimization algorithms for structural design optimization," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 327–333, 2013.
- [4] A. R. Yildiz and K. N. Solanki, "Multi-objective optimization of vehicle crashworthiness using a new particle swarm based approach," *International Journal of Advanced Manufacturing Technology*, vol. 59, pp. 367–376, 2012.
- [5] A. F. Ulusoy and F. Erbatur, "Discrete and continuous structural optimisation using evolution strategies," in *Proceedings of the 3rd International Conference on Engineering Computational Technology (ICECT '03)*, B. H. V. Topping and Z. Bittnar, Eds., pp. 163–164, Civil-Comp press, Edinburgh, UK, 2002.
- [6] A. R. Yildiz and K. Saitou, "Topology synthesis of multicomponent structural assemblies in continuum domains," *Journal of Mechanical Design*, vol. 133, no. 1, Article ID 011008, 2011.
- [7] N. Pholdee and S. Bureerat, "Performance enhancement of multiobjective evolutionary optimizers for truss design using an approximate gradient," *Computers and Structures*, vol. 106–107, pp. 115–124, 2012.
- [8] I. Durgun and A. R. Yildiz, "Structural design optimization of vehicle components using Cuckoo search algorithm," *Materials Testing*, vol. 54, no. 3, pp. 185–188, 2012.
- [9] S. Chen, Y. Zheng, C. Cattani, and W. Wang, "Modeling of biological intelligence for SCM system optimization," *Computational and Mathematical Methods in Medicine*, vol. 2012, Article ID 769702, 10 pages, 2012.
- [10] M. Zirour, C. Y. Liang, and W. R. Ismail, "Genetic algorithm for a quay management problem," *Australian Journal of Basic and Applied Sciences*, vol. 5, no. 9, pp. 2257–2263, 2011.
- [11] S. Y. Chen and Y. F. Li, "Automatic sensor placement for model-based robot vision," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 34, no. 1, pp. 393–408, 2004.
- [12] A. R. Yildiz, "A novel hybrid immune algorithm for global optimization in design and manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 2, pp. 261–270, 2009.
- [13] A. R. Yildiz, "A new hybrid bee colony optimization approach for robust optimal design and manufacturing," *Applied Soft Computing*. In press.
- [14] V. Kelner, F. Capitanescu, O. Léonard, and L. Wehenkel, "A hybrid optimization technique coupling an evolutionary and a local search algorithm," *Journal of Computational and Applied Mathematics*, vol. 215, no. 2, pp. 448–456, 2008.
- [15] A. R. Yildiz, "A new design optimization framework based on immune algorithm and Taguchi's method," *Computers in Industry*, vol. 60, no. 8, pp. 613–620, 2009.
- [16] S. Kanyakam and S. Bureerat, "Comparative performance of surrogate-assisted MOEAs for geometrical design of pin-fin heat sinks," *Applied Mathematics*, vol. 2012, Article ID 534783, 14 pages, 2012.
- [17] T. Kunakote and S. Bureerat, "Multi-objective topology optimization using evolutionary algorithms," *Engineering Optimization*, vol. 43, no. 5, pp. 541–557, 2011.

- [18] W. Achtziger, "On simultaneous optimization of truss geometry and topology," *Structural and Multidisciplinary Optimization*, vol. 33, no. 4-5, pp. 285-304, 2007.
- [19] W. Tang, L. Tong, and Y. Gu, "Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables," *International Journal for Numerical Methods in Engineering*, vol. 62, no. 13, pp. 1737-1762, 2005.
- [20] K. Shea and I. F. C. Smith, "Improving full-scale transmission tower design through topology and shape optimization," *Journal of Structural Engineering*, vol. 132, no. 5, pp. 781-790, 2006.
- [21] C. M. Chan and K. M. Wong, "Structural topology and element sizing design optimisation of tall steel frameworks using a hybrid OC-GA method," *Structural and Multidisciplinary Optimization*, vol. 35, no. 5, pp. 473-488, 2008.
- [22] J. Zhu, W. Zhang, P. Beckers, Y. Chen, and Z. Guo, "Simultaneous design of components layout and supporting structures using coupled shape and topology optimization technique," *Structural and Multidisciplinary Optimization*, vol. 36, no. 1, pp. 29-41, 2008.
- [23] M. Giger and P. Ermanni, "Evolutionary truss topology optimization using a graph-based parameterization concept," *Structural and Multidisciplinary Optimization*, vol. 32, no. 4, pp. 313-326, 2006.
- [24] R. Su, X. Wang, L. Gui, and Z. Fan, "Multi-objective topology and sizing optimization of truss structures based on adaptive multi-island search strategy," *Structural and Multidisciplinary Optimization*, vol. 43, no. 2, pp. 275-286, 2011.
- [25] V. Togan and A. Daloglu, "Optimization of 3d trusses with adaptive approach in genetic algorithms," *Engineering Structures*, vol. 28, pp. 1019-1027, 2006.
- [26] H. Rahami, A. Kaveh, and Y. Gholipour, "Sizing, geometry and topology optimization of trusses via force method and genetic algorithm," *Engineering Structures*, vol. 30, no. 9, pp. 2360-2369, 2008.
- [27] O. Hasanc and F. Erbatur, "Layout optimization of trusses using simulated annealing," *Advances in Engineering Software*, vol. 33, pp. 681-696, 2002.
- [28] C. Noilublao and S. Bureerat, "Simultaneous topology shape and sizing optimisation of skeletal structures using multiobjective evolutionary algorithms," in *Evolutionary Computation*, pp. 487-580, 2009.
- [29] T. Hagishita and M. Ohsaki, "Topology optimization of trusses by growing ground structure method," *Structural and Multidisciplinary Optimization*, vol. 37, no. 4, pp. 377-393, 2009.
- [30] A. Asadpoure, M. Tootkaboni, and J. K. Guest, "Robust topology optimization of structures with uncertainties in stiffness—application to truss structures," *Computers and Structures*, vol. 89, no. 11-12, pp. 1131-1141, 2011.
- [31] M. Beckers and C. Fleury, "A primal-dual approach in truss topology optimization," *Computers and Structures*, vol. 64, no. 1-4, pp. 77-88, 1997.
- [32] I. Hajirasouliha, K. Pilakoutas, and H. Moghaddam, "Topology optimization for the seismic design of truss-like structures," *Computers and Structures*, vol. 89, no. 7-8, pp. 702-711, 2011.
- [33] C. Noilublao and S. Bureerat, "Topology and sizing optimization of trusses with adaptive ground finite elements using multiobjective PBIL," *Advanced Materials Research*, vol. 308-310, pp. 1116-1121, 2011.
- [34] S. Silih, M. Premrov, and S. Kravanja, "Optimum design of plane timber trusses considering joint flexibility," *Engineering Structures*, vol. 27, no. 1, pp. 145-154, 2005.
- [35] L. Gil and A. Andreu, "Shape and cross-section optimization of a truss structure," *Computers and Structures*, vol. 79, no. 7, pp. 681-689, 2001.
- [36] R. Karkauskas and A. Norkus, "Truss optimization under stiffness, stability constraints and random loading," *Mechanics Research Communications*, vol. 33, no. 2, pp. 177-189, 2006.
- [37] A. Kaveh and S. Talatahari, "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures," *Computers and Structures*, vol. 87, no. 5-6, pp. 267-283, 2009.
- [38] A. C. C. Lemonge, H. J. C. Barbosa, A. L. G. A. Coutinho, and C. C. H. Borges, "Multiple cardinality constraints and automatic member grouping in the optimal design of steel framed structures," *Engineering Structures*, vol. 33, no. 2, pp. 433-444, 2011.
- [39] C. Noilublao and S. Bureerat, "Simultaneous topology and shape optimisation of a 3D framework structure using simulated annealing," in *Proceedings of the Technology and Innovation for Sustainable Development Conference (TISD '08)*, Faculty of Engineering, Khon Kaen University, January 2008.
- [40] P. Martínez, P. Martí, and O. M. Querin, "Growth method for size, topology, and geometry optimization of truss structures," *Structural and Multidisciplinary Optimization*, vol. 33, no. 1, pp. 13-26, 2007.
- [41] M. Zhou, N. Pagaldipti, H. L. Thomas, and Y. K. Shyy, "An integrated approach to topology, sizing, and shape optimization," *Structural and Multidisciplinary Optimization*, vol. 26, no. 5, pp. 308-317, 2004.
- [42] T. Sokol, "A 99 line code for discretized Michell truss optimization written in Mathematica," *Structural and Multidisciplinary Optimization*, vol. 43, no. 2, pp. 181-190, 2011.
- [43] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design, Optimization and Control*, Barcelona, Spain, 2002.
- [44] S. Bureerat and S. Srisomporn, "Optimum plate-fin heat sinks by using a multi-objective evolutionary algorithm," *Engineering Optimization*, vol. 42, no. 4, pp. 305-323, 2010.
- [45] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multiobjective optimization algorithm: AMOSA," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 269-283, 2008.
- [46] M. Reyes-Sierra and C. A. C. Coello, "Multi-objective particle swarm optimizers: a survey of the state-of-the-art," *International Journal of Computational Intelligence Research*, vol. 2, pp. 287-308, 2006.
- [47] T. Aittokoski and K. Miettinen, "Efficient evolutionary approach to approximate the Pareto-optimal set in multiobjective optimization, UPS-EMOA," *Optimization Methods and Software*, vol. 25, no. 6, pp. 841-858, 2010.
- [48] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, 1999.
- [49] W. A. Bennage and A. K. Dhingra, "Single and multiobjective structural optimization in discrete-continuous variables using simulated annealing," *International Journal for Numerical Methods in Engineering*, vol. 38, no. 16, pp. 2753-2773, 1995.
- [50] S. Bureerat and J. Limtragool, "Performance enhancement of evolutionary search for structural topology optimisation," *Finite Elements in Analysis and Design*, vol. 42, no. 6, pp. 547-566, 2006.

- [51] J. D. Knowles and D. W. Corne, "Approximating the non-dominated front using the Pareto archived evolution strategy," *Evolutionary computation*, vol. 8, no. 2, pp. 149–172, 2000.
- [52] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [53] K. Deb, A. Pratap, and T. Meyarivan, "Constrained test problems for multi-objective evolutionary optimization," KanGAL Report No. 200002, Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology, Kanpur, India, 2000.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

